

# Network Structures and Fast Distributed MMSE Estimation

Muhammed O. Sayin<sup>a,\*</sup>, Suleyman S. Kozat<sup>b</sup>

<sup>a</sup>The Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61801 USA

<sup>b</sup>The Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara 06800 Turkey, Tel: +90 (312) 290-2336, Fax: +90 (312) 290-1223

---

## Abstract

We construct optimal estimation algorithms over distributed networks for state estimation in the mean-square error (MSE) sense. Here, we have a distributed collection of agents with processing and cooperation capabilities. These agents continually observe a noisy version of a desired state of the nature through a linear model and seek to learn this state by interacting with each other. Although this problem has attracted significant attention and extensively been studied in several different fields including machine learning theory to signal processing, all the well-known strategies achieve suboptimal learning performance in the MSE sense. To this end, we provide algorithms that achieve distributed minimum MSE (MMSE) performance over an arbitrary network topology based on the aggregation of information at each agent. This approach differs from the diffusion of information across network, i.e., exchange of local estimates per time instance. Importantly, we show that exchange of local estimates is sufficient only over the certain network topologies. By inspecting these network structures, we also propose strategies that achieve the distributed MMSE performance also through the diffusion of information such that we can substantially reduce the communication load while achieving the best possible MSE performance. For practical implementations we provide approaches to reduce the complexity of the algorithms through the time-windowing of the observations. Finally, in the numerical examples, we demonstrate the superior performance of the introduced algorithms in the MSE sense due to optimal estimation.

**Keywords:** Distributed networks, distributed Kalman filter, MMSE estimation, tree networks.

---

## 1. Introduction

Over a distributed network of agents with measurement, processing and communication capabilities, we can have enhanced processing performance, e.g., fast response time, relative to the centralized networks by distributing the processing power over the networks [1, 2, 3]. Mainly, distributed agents observe a true state of the system through noisy measurements from different viewpoints, process the observation data in order to estimate the state, and communicate with each other to alleviate the estimation process in a fully distributed manner. Notably, the agents

---

\*Corresponding author

Email addresses: sayin2@illinois.edu (Muhammed O. Sayin), kozat@ee.bilkent.edu.tr (Suleyman S. Kozat)

Preprint submitted to Elsevier

October 4, 2016

can respond to streaming data in an online manner by disclosing information among each other at certain instances. This framework is conveniently used to model highly complex structures from defense applications to social and economical networks [4, 5, 6, 7]. As an example, say that we have radar systems distributed over an area and seeking to locate hostile missiles, i.e., the location of the missile is the underlying state of the system. In that respect, distributed processing approach has vital importance in terms of detecting the missiles and reacting as fast as possible. In particular, even if the viewpoints of a few radar systems are blocked due to environmental obstacles, through the communication among the radar systems, each system can still locate the missiles. Additionally, since each radar system not only collects measurements but also process them to locate the missiles, the overall system can react against the missiles faster than a centralized approach in which measurements of all the radar systems are collected at a centralized unit and processed together.

Although there is an extensive literature on this field, e.g., [2, 8, 9, 10, 5, 6, 7] and references therein, we still have significant and yet unexplored problems for disclosure and utilization of information among agents. Prior art focuses on the computationally simple algorithms that aim to achieve certain performance criterion asymptotically, e.g., diffusion or consensus based estimation algorithms [2, 8, 11, 12], due to processing power related practical concerns. However, there is a trade-off in terms of computational complexity and estimation performance. Correspondingly, for certain network scenarios, as exemplified in the following, computationally demanding yet fast algorithms can be preferable to achieve desired performance. In particular, commonly, at each iteration of the algorithms, agents make a noisy measurement, process the data, and exchange information with each other. In this chain of steps, each step, i.e., measurement, processing, and communication steps, plays significant roles for the overall performance of the algorithms as follows:

*Speed of reliable communication:* The reliable communication rate among the agents, i.e., communication step, should be as fast as the processing step for fast estimation performance. Otherwise, the communication rate creates a burden for the overall speed of the algorithm. In chemical kinetics, this phenomena is called as the rate determining step (RDS) such that speed of a chain of reactions can be approximately determined by the speed of the slowest reaction [13]. This implies that we can employ fast algorithms in spite of high complexity in the network scenarios with relatively slow communication step due to time flexibility.

*Communication power:* The low complexity algorithms have relatively slow learning rate such that agents disclose similar information with the previously disclosed ones. However, communication between the agents also requires significant power resources. To this end, in the design of distributed networks, we should also consider the trade-off in terms of the communication power and the estimation performance [4, 14, 12]. In [14, 12], the authors study this problem by compressing the data before the transmission and extracting the compressed information adaptively. We point out that in a fast estimation algorithm, exchanged data carries relatively more, i.e., new, information. This implies a communication flexibility such that through certain protocols, we can reduce the number of information exchange due to the disclosure of fertile information, and correspondingly this can reduce the communication load of the overall system.

*Number of samples:* Finally, in certain network scenarios, the number of measurements can also be too low, e.g., due to small sampling frequency, that the low complexity approaches aiming to achieve the performance asymptotically cannot yield an acceptable result. Particularly, some applications can require the agents to process the measurements as much as they can, i.e., optimally with respect to certain performance measure, due to the limited number of measurements.

As noted above, computationally demanding yet *fast* algorithms have applications for certain network scenarios. Hence, formulating the optimal distributed estimation algorithms with respect to certain performance criteria is a significant and unexplored challenge. To this end, instead of computationally simple yet slow algorithms, we seek to formulate the *optimal* estimation algorithms over distributed networks in the mean-square-error (MSE) sense. We design which information to disclose and how to utilize them for the minimum MSE (MMSE) performance. In addition to the aforementioned scenarios, we also analyze the network topologies in which the optimal algorithms can also be computationally simple and require less communication load. In the single agent systems, the well-known Kalman filter is extensively used for the MMSE estimation. Due to its iterative nature and superior estimation performance, Kalman filter appeals significant attention in both control theory and signal processing literatures. Correspondingly, there have been abundant attempts to formulate distributed Kalman filter [15, 16, 17, 18]. However, while extending Kalman filter for distributed networks, these approaches develop asymptotically optimal or sub-optimal approaches in the MSE sense due to practical concerns. Particularly, the cooperation among the agents results in complex MMSE algorithms for arbitrary network topologies. As explained in [19], once agent-1 transmits an information to agent-2, this information becomes common knowledge between the agents. This implies that agent-1 knows that agent-2 knows it, agent-2 knows that agent-1 knows it, agent-1 knows that agent-2 knows that agent-1 knows it, and so on. Therefore, in the design of the distributed MMSE estimator, we take these into consideration and utilize the exchanged information based on its content rather than a blind approach in which all exchanged information is handled irrespective of the content as in the diffusion or consensus based approaches.

In diffusion or consensus based approaches, the agents utilize the exchanged information generally through certain static combination rules, e.g., the uniform rule [20], the Laplacian rule [21] or the Metropolis rule [22]. However, if the statistical profile of the measurement data varies over the network, i.e., each agent observes diverse signal-to-noise ratios, by ignoring the variation in noise, these rules yield severe decline in the estimation performance [2]. In such cases the agents can even perform better without cooperation [2]. Hence, we also seek to formulate the optimal utilization of the exchanged information in the MSE sense.

Consider distributed networks of the agents that observe a noisy version of an underlying state and can exchange information with only certain agents at each time instant. Note that information is transmitted over the hops since certain agents might not be connected directly. To this end, we introduce a comprehensive cost measure considering the transmission of information over the hops and the corresponding content. We formulate the MMSE estimator, called the optimal distributed online learning (ODOL) algorithm, using common knowledge about the statistical profiles and the network topology for the jointly Gaussian state and noise signals. We point out that the ODOL algorithm achieves the linear MMSE (LMMSE) performance for other statistical profiles. Notably, the ODOL algorithm utilizes the aggregation of information at each agent via time-stamped information exchange and is different from the conventional algorithms that benefit from the diffusion of local estimates [2, 8, 11]. The aggregation of information at each agent requires excessive communication load relative to the diffusion of information in general. In particular, the ODOL algorithm is not practical for real life applications due to the excessive communication load, however, we utilize this algorithm to analyze the performance of proposed approaches since the ODOL algorithm achieves the oracle performance, i.e., the distributed MMSE performance, for any network scenario.

Furthermore, we modify the ODOL algorithm by exploiting the network topology to reduce the communication load such that agents disclose their local estimates only yet achieve the

MMSE performance. Over tree networks excluding multiple transmission paths, the diffusion of local estimates is sufficient to achieve the distributed MMSE performance. We analytically show that for sufficiency of diffusion of estimates, tree networks could also involve cell structures that we define as the sub-networks in which all agents are connected with each other. Additionally, we formulate the optimal and efficient distributed online learning (OEDOL) algorithm, which is practical for real life applications and achieves the MMSE performance over the tree networks. Note that for an arbitrary network topology, we can construct such network connections by eliminating certain communication links. Through numerical examples, we examine the impact of constructing the spanning tree of a network on the estimation performance and observe that the impact is negligible. Finally, we propose the time windowing of the observation set in order to reduce the complexity of the algorithms. In these algorithms, agents combine the received information linearly with time-invariant combination weights.

We can list our main contributions as follows: 1) We introduce a thorough cost measure considering the transmission of information over hops across networks. 2) We derive the ODOL algorithm achieving the oracle performance over arbitrary networks through the aggregation of information at each agent. 3) For practical applications, we study the network structures in which we can achieve the MMSE performance through the disclosure of local estimates. 4) We propose the OEDOL algorithm achieving the MMSE performance over certain network topologies with tremendously reduced communication load. 5) We also formulate sub-optimal versions of the algorithms with reduced complexity. 6) We provide numerical examples demonstrating the significant gains due to the introduced algorithms.

The remainder of the paper is organized as follows. We introduce the distributed MMSE framework in Section II. We study the tree networks, exploit the network topology to formulate the OEDOL algorithm that reduces the communication load and introduce cell structures enhancing the learning rate further under communication constraints in Section III. We propose the sub-optimal versions of the ODOL and OEDOL algorithms for practical implementations in Section IV. In Section V, we provide numerical examples demonstrating significant gains due to the introduced algorithms compared to the conventional algorithms. We conclude the paper in Section VI with several remarks.

*Notation:* Bold lower (or upper) case letters denote column vectors (or matrices). For a vector  $\mathbf{a}$  (or matrix  $\mathbf{A}$ ),  $\mathbf{a}^T$  (or  $\mathbf{A}^T$ ) is its ordinary transpose. The terms of the vector  $\mathbf{1}$  (and  $\mathbf{0}$ ) are all 1s (and 0s) and the size of the vector will be understood from the context. We use calligraphic letters for random variables and underlined calligraphic letters for random vectors, e.g.,  $\mathcal{X}$  and  $\underline{\mathcal{X}}$ . Bold calligraphic letters, e.g.,  $\mathcal{Z}$ , denote a set of random variables. For a random variable  $X$  (or vector  $\underline{X}$ ),  $E[X]$  (or  $E[\underline{X}]$ ) represents its expectation. We work with real data for notational simplicity. The operator  $\text{col}\{\cdot\}$  produces a column vector or a matrix in which the arguments of  $\text{col}\{\cdot\}$  are stacked one under the other. For a matrix argument,  $\text{diag}\{\mathbf{A}\}$  operator constructs a diagonal matrix with the diagonal entries of  $\mathbf{A}$ . For a given set  $N$ ,  $\text{diag}\{N\}$  creates a diagonal matrix whose diagonal block entries are elements of the set. The operator  $\otimes$  denotes the Kronecker product.

## 2. Distributed-MMSE Estimation Framework

Consider a distributed network of  $m$  agents with processing and communication capabilities. In Fig. 1, we illustrate this network through an undirected graph, where the vertices and the edges correspond to the agents and the communication links across the network, respectively.

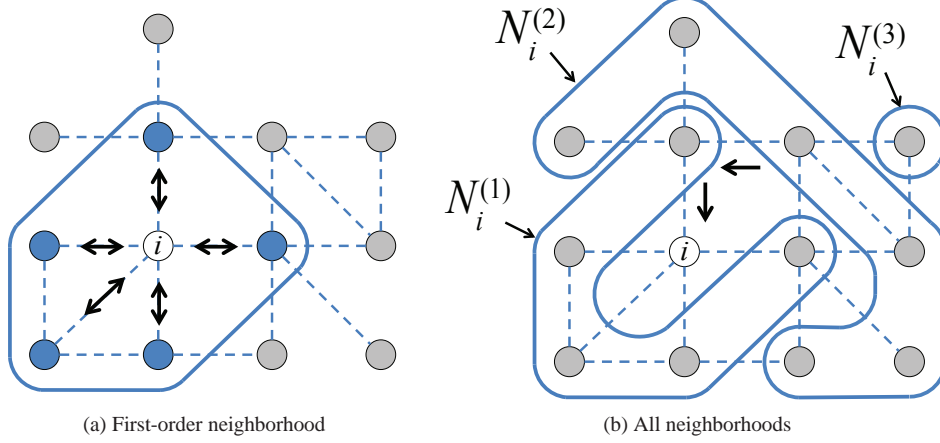


Figure 1: The neighborhoods of  $i$ th agent over the distributed network.

For each agent  $i$ , we denote the set of agents whose information could be received at least after  $k$  hops, i.e.,  $k$ -hop neighbors, by  $N_i^{(k)}$  defined as

$$N_i^{(k)} \triangleq \{j_1, \dots, j_{\pi_i^{(k)}}\}$$

and  $\pi_i^{(k)} = |N_i^{(k)}|$  is the cardinality of  $N_i^{(k)}$  (See Fig. 1b). We assume that  $N_i^{(0)} = \{i\}$  and  $N_i^{(k)} = \emptyset$  for  $k < 0$ . Here, the agents observe a noisy version of a time-invariant and unknown state vector  $\mathbf{x} \in \mathbb{R}^p$  which is a realization of the Gaussian random vector process  $\underline{\mathbf{X}}$  with mean  $\bar{\mathbf{x}}$  and auto-covariance  $\Sigma_{\mathbf{x}}$ . Each agent  $i$  observes the state as

$$\mathbf{y}_{i,t} = \mathbf{H}_i \mathbf{x} + \mathbf{n}_{i,t},$$

where  $\mathbf{H}_i \in \mathbb{R}^{q \times p}$  is a known matrix,  $\mathbf{n}_{i,t} \in \mathbb{R}^q$  is a realization of a zero-mean white Gaussian vector process  $\underline{\mathbf{n}}_{i,t}$  with auto-covariance  $\Sigma_{\mathbf{n}_i}$  and correspondingly the observation  $\mathbf{y}_{i,t} \in \mathbb{R}^q$  is a realization of the random process  $\underline{\mathbf{y}}_{i,t} = \mathbf{H}_i \underline{\mathbf{X}} + \underline{\mathbf{n}}_{i,t}$ . The noise  $\mathbf{n}_{i,t}$  is also independent from the state and the other noise parameters.

We assume that the variance of the noise signals are known, which can also be readily estimated from the data [23]. At each instant, the agents observe the underlying state through  $\mathbf{y}_{i,t}$ , diffuse information to the neighboring agents and receive information from them as seen in Fig. 1a. Naturally, the agents can learn the true state, under certain regularity conditions [23], irrespective of cooperation among them, provided that their own measurements are not biased and sufficient to estimate the true state in time. However, through the cooperation of agents, we can increase the learning rate significantly [2]. To this end, we aim to find the optimal learning strategy over distributed networks in the MSE sense.

As a *lower bound* on the MSE performance over distributed networks, we first consider the case when there is no limitation on the amount of the disclosed information. Agents disclose their own measurements and transmit the received information from the neighboring agents (with time stamps, i.e., the information is ordered). Through this implementation, each agent can obtain the measurements of the other agents separately in a connected network. We point

out that the information on the non-neighboring agents could only be received after certain hops due to the sparsely connected structure. As an example, the disclosed information of  $j \in N_i^{(2)}$  reaches to  $i$  by passing through two communication links as seen in Fig. 1b. In particular, this case assumes that each agent has access to full information from the other agents, albeit with certain hops, and corresponds to the direct aggregation of all information across the network at each agent.

**Remark 2.1:** The aggregation of all information directly at each agent and processing them optimally yield excessive communication load in general. Since the communication load is crucial for the applicability of the distributed learning algorithms [12, 14], we also seek to reduce the communication load yet achieve the MMSE performance. In the consensus and diffusion implementations, the agents disclose the local estimates, which requires relatively reduced communication load [2, 24, 11, 8]. However, later in this paper we show that the disclosure of the local estimates is sufficient to achieve the MMSE performance *only* over certain network topologies.

At time  $t$ , all the information aggregated at  $i$ th agent is given by

$$\left\{ \{y_{i,\tau}\}_{\tau \leq t}, \{y_{j,\tau}\}_{j \in N_i^{(1)}, \tau \leq t-1}, \dots, \{y_{j,\tau}\}_{j \in N_i^{(\kappa_i)}, \tau \leq t-\kappa_i} \right\},$$

where the information from the furthest agent is received at least after  $\kappa_i$  hops<sup>1</sup>. Here, the collection set  $\{y_{j,\tau}\}_{j \in N_i^{(k)}, \tau \leq t-k}$  includes the observations from the  $k$ -hop neighbors  $N_i^{(k)}$ . Particularly, the collection is explicitly defined as

$$\begin{aligned} \{y_{j,\tau}\}_{j \in N_i^{(k)}, \tau \leq t-k} \triangleq & \{y_{j_1, t-k}, \dots, y_{j_1, 0}, \\ & \dots, y_{j_{\pi_i(k)}, t-k}, \dots, y_{j_{\pi_i(k)}, 0}\}. \end{aligned} \quad (1)$$

Then, we have the comprehensive cost measure in the distributed-MMSE framework as

$$\begin{aligned} \hat{\mathbf{x}}_{i,t} = \min_{\mathbf{x}} E \left[ \left\| \underline{\mathbf{X}}_t - \mathbf{x} \right\|^2 \middle| \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}_{\tau \leq t}, \right. \\ \left. \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(1)}, \tau \leq t-1}, \dots, \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(\kappa_i)}, \tau \leq t-\kappa_i} \right] \end{aligned} \quad (2)$$

and the MMSE estimate for each agent  $i$  is given by the expectation of  $\mathbf{x}$  conditioned all the accessed information:

$$\begin{aligned} \mathbf{x}_{i,t} = E \left[ \underline{\mathbf{X}}_t \middle| \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}_{\tau \leq t}, \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(1)}, \tau \leq t-1}, \dots, \right. \\ \left. \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(\kappa_i)}, \tau \leq t-\kappa_i} \right]. \end{aligned} \quad (3)$$

Correspondingly, we define the random variable  $\hat{\underline{\mathbf{X}}}_{i,t}$  as follows

$$\hat{\underline{\mathbf{X}}}_{i,t} \triangleq E \left[ \underline{\mathbf{X}}_t \middle| \left\{ \underline{\mathbf{y}}_{i,\tau} \right\}_{\tau \leq t}, \dots, \left\{ \underline{\mathbf{y}}_{j,\tau} \right\}_{j \in N_i^{(\kappa_i)}, \tau \leq t-\kappa_i} \right].$$

---

<sup>1</sup>For notational simplicity, we denote  $N_i \triangleq N_i^{(1)}$  and  $\pi_i \triangleq \pi_i^{(1)}$ .

### 2.1. ODOL Algorithm

Importantly, since the state and the observation noise are independent Gaussian random parameters, we propose the ODOL algorithm constructing the distributed-MMSE estimator (3) in an iterative way. To this end, we collect all received information at time  $t$  as

$$\mathbf{z}_{i,t} = \text{col}\{\mathbf{y}_{i,t}, \mathbf{y}_{i_1,t-1}, \dots, \mathbf{y}_{i_{\pi_i},t-1}, \dots, \mathbf{y}_{j_1,t-K_i}, \dots, \mathbf{y}_{j_{\pi_i},t-K_i}\}.$$

Then, the iterations of the ODOL algorithm are given by

$$\begin{aligned} \mathbf{K}_{i,t} &= \hat{\Sigma}_{i,t-1} \bar{\mathbf{H}}_i^T (\bar{\mathbf{H}}_i \hat{\Sigma}_{i,t-1} \bar{\mathbf{H}}_i^T + \bar{\Sigma}_{n_i})^{-1}, \\ \hat{\mathbf{x}}_{i,t} &= (\mathbf{I} - \mathbf{K}_{i,t} \bar{\mathbf{H}}_i) \hat{\mathbf{x}}_{i,t-1} + \mathbf{K}_{i,t} \mathbf{z}_{i,t}, \\ \hat{\Sigma}_{i,t} &= (\mathbf{I} - \mathbf{K}_{i,t} \bar{\mathbf{H}}_i) \hat{\Sigma}_{i,t-1}, \end{aligned}$$

where<sup>2</sup>  $\bar{\mathbf{H}}_i \triangleq (\mathbf{P}_i \otimes \mathbf{I}_q) \mathbf{H}$ ,  $\mathbf{H} \triangleq \text{col}\{\mathbf{H}_1, \dots, \mathbf{H}_m\}$ ,  $\bar{\Sigma}_{n_i} \triangleq (\mathbf{P}_i \otimes \mathbf{I}_p) \Sigma_n (\mathbf{P}_i \otimes \mathbf{I}_p)^T$ ,  $\Sigma_n \triangleq \text{diag}\{\Sigma_{n_1}, \dots, \Sigma_{n_m}\}$ , and  $\mathbf{P}_i$  is the corresponding permutation matrix.

### 3. Distributed-MMSE Estimation with Disclosure of Local Estimate

Since the underlying state and the observation noises are independent Gaussian random parameters, the conditional expectation of the state given the observations, i.e., the MMSE estimate, is an affine combination of the observations. Hence, in this section we aim to exploit the affine transformation in order to reduce the disclosed amount of information so that each agent exchanges the necessary information in an efficient way. We point out that, as shown in the following example, disclosure of the local estimates is not sufficient to achieve the distributed MMSE performance in general. Consider a cycle network of 4 agents where agent-1 is directly connected with agents 2 and 3. At time  $t = 1$ , we obtain

$$\begin{aligned} \hat{\mathbf{x}}_{2,1} &= E[\mathcal{X} | \underline{\mathcal{Y}}_{2,1} = \mathbf{y}_{2,1}, \underline{\mathcal{Y}}_{2,0} = \mathbf{y}_{2,0}, \underline{\mathcal{Y}}_{1,0} = \mathbf{y}_{1,0}, \underline{\mathcal{Y}}_{4,0} = \mathbf{y}_{4,0}], \\ \hat{\mathbf{x}}_{3,1} &= E[\mathcal{X} | \underline{\mathcal{Y}}_{3,1} = \mathbf{y}_{3,1}, \underline{\mathcal{Y}}_{3,0} = \mathbf{y}_{3,0}, \underline{\mathcal{Y}}_{1,0} = \mathbf{y}_{1,0}, \underline{\mathcal{Y}}_{4,0} = \mathbf{y}_{4,0}]. \end{aligned}$$

Since  $\hat{\mathbf{x}}_{2,1}$  and  $\hat{\mathbf{x}}_{3,1}$  are conditioned on  $\underline{\mathcal{Y}}_{4,0} = \mathbf{y}_{4,0}$  commonly, the MMSE estimator  $\hat{\mathbf{x}}_{1,2}$  could not be obtained by just processing the current estimates, i.e.,

$$\begin{aligned} \hat{\mathbf{x}}_{1,2} &\neq E[\mathcal{X} | \underline{\mathcal{Y}}_{1,2} = \mathbf{y}_{1,2}, \hat{\mathcal{X}}_{1,1} = \hat{\mathbf{x}}_{1,1}, \hat{\mathcal{X}}_{2,1} = \hat{\mathbf{x}}_{2,1}, \\ &\quad \hat{\mathcal{X}}_{3,1} = \hat{\mathbf{x}}_{3,1}]. \end{aligned}$$

In particular, for i.i.d. observations  $\underline{\mathcal{Y}}_1$ ,  $\underline{\mathcal{Y}}_2$  and  $\underline{\mathcal{Y}}_3$  we have

$$E[\mathcal{X} | \underline{\mathcal{Y}}_1, \underline{\mathcal{Y}}_2, \underline{\mathcal{Y}}_3] \neq E[\mathcal{X} | E[\mathcal{X} | \underline{\mathcal{Y}}_1, \underline{\mathcal{Y}}_2], E[\mathcal{X} | \underline{\mathcal{Y}}_2, \underline{\mathcal{Y}}_3]].$$

Hence, optimal processing of estimates should be elaborately considered. In the following, we analytically show that the MMSE estimate could be obtained through the disclosure of local estimates over “tree networks” but not in general.

<sup>2</sup>If the inverse fails to exist, a pseudo inverse can replace the inverse [25].



### 3.1. Tree Networks

A network has a “tree structure” if its corresponding graph is a *tree*, i.e., connected and undirected without any cycles [26]. As an example, the conventional star or line networks have tree structures. We remark that for an arbitrary network topology we can also construct the spanning tree of the network and eliminate the cycles. In the literature, there exists numerous distributed algorithms for minimum spanning tree construction [27, 28, 29, 30, 31].

Importantly, the following theorem shows that over tree networks we can achieve the performance of the oracle algorithm through the disclosure of the local estimates only.

**Theorem 3.1:** *Consider the distributed estimation framework over a tree network. Then, the distributed-MMSE estimator (3) could also be obtained by*

$$\hat{\mathbf{x}}_{i,t} = E \left[ \underline{X} \left| \left\{ \underline{y}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq t}, \left\{ \hat{\underline{x}}_{j,\tau} = \hat{\mathbf{x}}_{j,\tau} \right\}_{j \in N_i}^{\tau \leq t-1} \right. \right], \quad (4)$$

i.e., each agent can only disclose its own local estimate to achieve the MMSE performance.

*Proof:* Initially, agent  $i$  has access to  $\mathbf{y}_{i,0}$  only and the MMSE estimator is  $\hat{\mathbf{x}}_{i,0} = E[\underline{X}|\underline{y}_{i,0} = \mathbf{y}_{i,0}]$ . At time  $t = 1$ , the MMSE estimator is given by

$$\hat{\mathbf{x}}_{i,1} = E \left[ \underline{X} \left| \left\{ \underline{y}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq 1}, \left\{ \underline{y}_{j,0} = \mathbf{y}_{j,0} \right\}_{j \in N_i} \right. \right], \quad (5)$$

which can be written as

$$\begin{aligned} \hat{\mathbf{x}}_{i,1} &= E \left[ \underline{X} \left| \left\{ \underline{y}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq 1}, \left\{ E[\underline{X}|\underline{y}_{j,0} = \mathbf{y}_{j,0}] \right\}_{j \in N_i} \right. \right], \\ &= E \left[ \underline{X} \left| \left\{ \underline{y}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq 1}, \left\{ \hat{\underline{x}}_{j,0} = \hat{\mathbf{x}}_{j,0} \right\}_{j \in N_i} \right. \right]. \end{aligned} \quad (6)$$

As seen in Fig. 2, over a tree network, for  $k \in \{1, \dots, \kappa_i\}$  we have

$$N_i^{(k)} = \bigcup_{j \in N_i} \left( N_i^{(k)} \cap N_j^{(k-1)} \right). \quad (7)$$

Note that the sets in (7) are disjoint as

$$\left( N_i^{(k)} \cap N_{j_1}^{(k-1)} \right) \cap \left( N_i^{(k)} \cap N_{j_2}^{(k-1)} \right) = \emptyset \quad (8)$$

for all  $j_1, j_2 \in N_i$  and  $j_1 \neq j_2$ . Notably, over a tree network, by (8), we can partition the collection set defined in (1) as follows

$$\begin{aligned} \left\{ \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(k)}} &= \left\{ \left\{ \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(k)} \cap N_{j_1}^{(k-1)}}, \right. \\ &\quad \left. \dots, \left\{ \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(k)} \cap N_{j_{\kappa_i}}^{(k-1)}} \right\}. \end{aligned} \quad (9)$$

Let the set of all accessed information by the  $i$ th agent at time  $t = 2$  be

$$Z_{i,2} \triangleq \left\{ \left\{ \mathbf{y}_{i,\tau} \right\}^{\tau \leq 2}, \left\{ \mathbf{y}_{j,\tau} \right\}_{j \in N_i}^{\tau \leq 1}, \left\{ \mathbf{y}_{k,0} \right\}_{k \in N_i^{(2)}} \right\}.$$



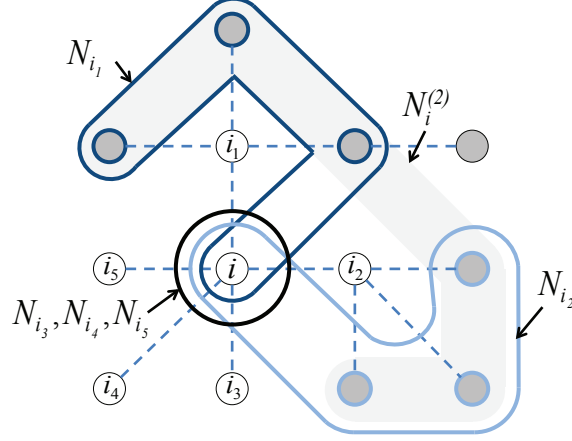


Figure 2: An example tree network. Notice the eliminated links from Fig. 1 to avoid multi path information diffusion.

We also define

$$\hat{Z}_{j,i,1} \triangleq \left\{ \overbrace{\{\mathbf{y}_{k,1}\}_{k \in N_i^{(1)} \cap N_j^{(0)}}}^{=\mathbf{y}_{j,1}}, \{\mathbf{y}_{k,0}\}_{k \in N_i^{(2)} \cap N_j^{(1)}} \right\}$$

such that by (9) we have

$$Z_{i,2} = \{\mathbf{y}_{i,2}, \hat{Z}_{j_1,i,1}, \dots, \hat{Z}_{j_{n_i},i,1}, Z_{i,1}\} \quad (10)$$

and

$$\hat{Z}_{j,i,1} = Z_{j,1} \setminus \left\{ \overbrace{\mathbf{y}_{j,0}}^{=Z_{j,0}}, \mathbf{y}_{i,0} \right\}. \quad (11)$$

Hence, by (10) and (11), we obtain

$$\begin{aligned} \hat{\mathbf{x}}_{i,2} &= E \left[ \underline{X} \mid \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq 2}, \left\{ \underline{\mathbf{z}}_{j,0} = Z_{j,0} \right\}_{j \in N_i} \right], \\ \left\{ \underline{\mathbf{z}}_{j,1} \setminus \left\{ \underline{\mathbf{z}}_{j,0}, \underline{\mathbf{y}}_{i,0} \right\} = Z_{j,1} \setminus \left\{ Z_{j,0}, \mathbf{y}_{i,0} \right\} \right\}_{j \in N_i} &. \end{aligned} \quad (12)$$

After some algebra, (12) could also be obtained by

$$\begin{aligned} \hat{\mathbf{x}}_{i,2} &= E \left[ \underline{X} \mid \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq 2}, \left\{ E[\underline{X} \mid \underline{\mathbf{z}}_{j,\tau} = Z_{j,\tau}] \right\}_{j \in N_i}^{\tau \leq 1} \right], \\ &= E \left[ \underline{X} \mid \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq 2}, \left\{ \hat{\underline{\mathbf{x}}}_{j,\tau} = \hat{\mathbf{x}}_{j,\tau} \right\}_{j \in N_i}^{\tau \leq 1} \right]. \end{aligned} \quad (13)$$

At time  $t = 3$ , (10) yields

$$\hat{Z}_{j,i,2} = Z_{j,2} \setminus \left\{ Z_{j,1} \cup \overbrace{\left\{ Z_{i,1} \setminus \left\{ Z_{i,0} \cup Z_{j,0} \right\} \right\}}^{=\hat{Z}_{i,j,1}} \right\}. \quad (14)$$

Correspondingly, (14) leads to

$$\hat{Z}_{j,i,t} = Z_{j,t} \setminus \{Z_{j,t-1} \cup \hat{Z}_{i,j,t-1}\}$$

implying that  $\hat{Z}_{j,i,t}$  is constructible from  $Z_{i,\tau}$  and  $Z_{j,\tau}$  for  $\tau \leq t$ . Hence, for  $t > 0$  the MMSE estimator over tree networks is given by (4) and the proof is concluded.  $\square$

**Remark 3.1:** When the expectation of the state is conditioned on the infinite number of observations over even a constructed spanning tree, only a finite number of the observations is devoid compared to the case over a fully connected network. Hence, even if we construct the spanning tree of that network, we would still achieve the distributed MMSE performance over a fully connected (or centralized) network asymptotically. As an illustrative example, in Fig. 8, we observe that the MMSE performance over the fully connected, star and line networks are asymptotically the same. Similarly, in [32, 33, 34], the authors show that the performance of the diffusion based algorithms could approach the performance of a fully connected network under certain regularity conditions.

In the sequel, we propose the optimal and efficient distributed online learning (OEDOL) algorithm that achieves the distributed-MMSE performance over tree networks iteratively.

### 3.2. OEDOL Algorithm

We remark that the MMSE estimate  $\hat{\mathbf{x}}_{i,t}$  linearly depends on previous estimate  $\hat{\mathbf{x}}_{i,t-1}$ . In order to extract the new information, we need to eliminate the previously received information at each instant on the neighboring agents. This brings in additional computational complexity. On the contrary, agents can just disclose the new information, i.e.,  $\hat{\mathbf{z}}_{i,t}$ , after eliminating all the previously exchanged information. Since we are conditioning on the linear combinations of the conditioned variables without effecting their spanned space, i.e.,  $\hat{\mathbf{z}}_{i,t}$  is computable from  $\hat{\mathbf{x}}_{i,\tau}$  for  $\tau \leq t$  and vice versa, we can still achieve the MMSE performance by reduced computational load, yet.

At time  $t$ , agent  $i$  observes  $\mathbf{y}_{i,t}$  and receives  $\bar{\mathbf{z}}_{i,t} = \text{col}\{\hat{\mathbf{z}}_{j_1,t-1}, \dots, \hat{\mathbf{z}}_{j_{n_i},t-1}\}$ . Here, we determine the content of the received information to extract the innovation within them and utilize this innovation in the update of the local estimate. In particular, the OEDOL algorithm is given by

$$\hat{\mathbf{x}}_{i,t} = \mathbf{A}_{i,t}\hat{\mathbf{x}}_{i,t-1} + \mathbf{B}_{i,t}\mathbf{y}_{i,t} + \mathbf{C}_{i,t}\mathbf{w}_{i,t-1}, \quad (15)$$

$$\hat{\Sigma}_{i,t} = \mathbf{A}_{i,t}\hat{\Sigma}_{i,t-1}, \quad (16)$$

where  $\mathbf{w}_{i,t-1}$  is the innovation term extracted from the received information.  $\mathbf{A}_{i,t}$ ,  $\mathbf{B}_{i,t}$ , and  $\mathbf{C}_{i,t}$  are the corresponding weighting matrices defined as

$$\begin{aligned} \begin{bmatrix} \mathbf{B}_{i,t} & \mathbf{C}_{i,t} \end{bmatrix} &= \hat{\Sigma}_{i,t-1} \tilde{\mathbf{H}}_{i,t-1}^T \times \\ &\quad \left( \tilde{\mathbf{H}}_{i,t-1} \hat{\Sigma}_{i,t-1} \tilde{\mathbf{H}}_{i,t-1}^T + \tilde{\mathbf{G}}_{i,t-1} \right)^{-1}, \end{aligned} \quad (17)$$

$$\mathbf{A}_{i,t} = \mathbf{I} - \mathbf{B}_{i,t}\mathbf{H}_i - \mathbf{C}_{i,t}\tilde{\mathbf{H}}_{i,t-1}, \quad (18)$$

where  $\tilde{\mathbf{H}}_{i,t} = \text{col}\{\mathbf{H}_i, \bar{\mathbf{H}}_{i,t}\}$ ,  $\tilde{\mathbf{G}}_{i,t} = \text{diag}\{\Sigma_{n_i}, \bar{\mathbf{G}}_{i,t}\}$  and  $\bar{\mathbf{G}}_{i,t} = \text{diag}\{\mathbf{G}_{i,t}\}$ . The intermediate parame-

ters  $\bar{\mathbf{H}}_{i,t}$  and  $\mathbf{G}_{i,t}$  evolve according to

$$\bar{\mathbf{H}}_{i,t} = \begin{bmatrix} \mathbf{B}_{j_1,t} \mathbf{H}_{j_1} + \mathbf{C}_{j_1,t} \bar{\mathbf{H}}_{j_1,t-1} \\ \vdots \\ \mathbf{B}_{j_{\pi_i},t} \mathbf{H}_{j_{\pi_i}} + \mathbf{C}_{j_{\pi_i},t} \bar{\mathbf{H}}_{j_{\pi_i},t-1} \end{bmatrix} - \begin{bmatrix} \mathbf{C}_{j_1,t} \bar{\mathbf{H}}_{j_1,t-1}^{(i)} \\ \vdots \\ \mathbf{C}_{j_{\pi_i},t} \bar{\mathbf{H}}_{j_{\pi_i},t-1}^{(i)} \end{bmatrix}, \quad (19)$$

$$\mathbf{G}_{i,t} = \begin{bmatrix} \mathbf{B}_{j_1,t} \boldsymbol{\Sigma}_{n_{j_1}} \mathbf{B}_{j_1,t}^T + \mathbf{C}_{j_1,t} \bar{\mathbf{G}}_{j_1,t-1} \mathbf{C}_{j_1,t}^T \\ \vdots \\ \mathbf{B}_{j_{\pi_i},t} \boldsymbol{\Sigma}_{n_{j_{\pi_i}}} \mathbf{B}_{j_{\pi_i},t}^T + \mathbf{C}_{j_{\pi_i},t} \bar{\mathbf{G}}_{j_{\pi_i},t-1} \mathbf{C}_{j_{\pi_i},t}^T \end{bmatrix} - \begin{bmatrix} \mathbf{C}_{j_1,t}^{(i)} \mathbf{G}_{j_1,t-1}^{(i)} (\mathbf{C}_{j_1,t}^{(i)})^T \\ \vdots \\ \mathbf{C}_{j_{\pi_i},t}^{(i)} \mathbf{G}_{j_{\pi_i},t-1}^{(i)} (\mathbf{C}_{j_{\pi_i},t}^{(i)})^T \end{bmatrix} \quad (20)$$

and we initialize the parameters as  $\bar{\mathbf{H}}_{j,\tau} = \mathbf{0}$  and  $\mathbf{G}_{j,\tau} = \mathbf{0}$  for  $\tau < 0$ .

The recursion of the innovation parameter  $\mathbf{w}_{i,t}$  is given by

$$\mathbf{w}_{i,t} = \bar{\mathbf{z}}_{i,t} - \mathbf{D}_{i,t} \hat{\mathbf{z}}_{i,t-1} + \mathbf{T}_{i,t} \mathbf{w}_{i,t-2}, \quad (21)$$

where

$$\mathbf{D}_{i,t} = \text{col}\{\mathbf{C}_{j_1,t}^{(i)}, \dots, \mathbf{C}_{j_{\pi_i},t}^{(i)}\} \quad (22)$$

and

$$\mathbf{T}_{i,t} \triangleq \begin{bmatrix} \mathbf{C}_{j_1,t}^{(i)} \mathbf{C}_{i,t-1}^{(j_1)} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{C}_{j_{\pi_i},t}^{(i)} \mathbf{C}_{i,t-1}^{(j_{\pi_i})} \end{bmatrix}. \quad (23)$$

Then, the agents disclose

$$\hat{\mathbf{z}}_{i,t} = \hat{\mathbf{x}}_{i,t} - \mathbf{A}_{i,t} \hat{\mathbf{x}}_{i,t-1}.$$

The detailed description of the algorithm is provided in Table 1.

**Remark 3.2:** In (15), the combination matrices  $\mathbf{A}_{i,t}$ ,  $\mathbf{B}_{i,t}$  and  $\mathbf{C}_{i,t}$  are independent from the streaming data although they are time-varying. Hence they can be computed before-hand. In that case, as an example for  $p = q = 1$ , the computational complexity of the iterations is  $O(\bar{\pi}^2)$  where  $\bar{\pi}$  is the average of the cardinalities of the first-order neighborhoods across the network. Otherwise, the computational complexity of the algorithm is given by  $O(m(\bar{\pi})^3)$ , while it is  $O(m^3)$  for the oracle algorithm. Note that over the tree we have  $m - 1$  edges and correspondingly  $\bar{\pi}$  is expected to be small. Hence, the optimal diffusion strategy over tree networks also reduces the computational complexity in general (in addition to the substantial reduction in communication). In Table 2, we tabulate the computational complexities of the introduced algorithms.

While constructing the spanning tree, we cancel certain communication links in order to avoid multi-path information propagation. However, we also observe that in a fully connected network we can achieve the performance of the ODOL algorithm, i.e., the distributed-MMSE performance, through the disclosure of local observations. In particular, since all of the agents are connected, each agent can receive the observations across the network directly. Correspondingly, in a fully connected network, we can achieve identical performance with the

Table 1: The description of the OEDOL algorithm.

---

<b>Algorithm:</b> The OEDOL Algorithm
<b>Initialization:</b>
For $i = 1$ to $m$ do
$\hat{\mathbf{x}}_{i,-1} = \bar{\mathbf{x}}, \hat{\Sigma}_{i,-1} = \Sigma_x,$
$\tilde{\mathbf{H}}_{i,\tau} = \mathbf{0}, \mathbf{G}_{i,\tau} = \mathbf{0},$ and $\mathbf{w}_{i,t} = \mathbf{0}$ for $t < 0$
End for
<b>Iterations:</b>
Do for $t \geq 0$
For $i = 1$ to $m$ do
<b>Construction of Weights:</b>
For $j = 1$ to $m$ do
Calculate $\tilde{\mathbf{H}}_{j,t}$ and $\mathbf{G}_{j,t}$ by (19) and (20).
Determine combination matrices via (17), (18), (22), and (23).
End for
Construct $\bar{\mathbf{z}}_{i,t-1}$ through received $\hat{\mathbf{z}}_{j,t-1}$ for $j \in N_i$
<b>Extraction of Innovation:</b>
$\mathbf{w}_{i,t-1} = \bar{\mathbf{z}}_{i,t-1} - \mathbf{D}_{i,t-1}\hat{\mathbf{z}}_{i,t-2} + \mathbf{T}_{i,t-1}\mathbf{w}_{i,t-3}$
<b>Update:</b>
$\hat{\mathbf{x}}_{i,t} = \mathbf{A}_{i,t}\hat{\mathbf{x}}_{i,t-1} + \mathbf{B}_{i,t}\mathbf{y}_{i,t} + \mathbf{C}_{i,t}\mathbf{w}_{i,t-1}$
Disclose $\hat{\mathbf{z}}_{i,t} = \hat{\mathbf{x}}_{i,t} - \mathbf{A}_{i,t}\hat{\mathbf{x}}_{i,t-1}$ to the neighbors.
End for

---

Table 2: A comparison of the computational complexities of the proposed algorithms.

Algorithm	Without weights	Pre-computed weights
ODOL algorithm	$O(m^3)$	$O(m^2)$
OEDOL algorithm	$O(m(\bar{\pi})^3)$	$O(\bar{\pi}^2)$

ODOL algorithm only through the disclosure of the local estimates as stated in the following corollary formally.

**Corollary 3.1:** *Consider the distributed estimation framework over a fully connected network. Then, the distributed-MMSE estimator (3) could also be obtained by (4), where agents disclose local estimates only.*

*Proof:* Over a fully connected network,  $\kappa_i = 1$  and the MMSE estimator is given by

$$\hat{\mathbf{x}}_{i,t} = E \left[ \mathbf{X} \middle| \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq t}, \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i}^{\tau \leq t-1} \right] \quad (24)$$

and we can also obtain (24) by (6). Here, we have

$$\hat{Z}_{j,i,t} = Z_{j,t} \setminus \{Z_{i,t} \setminus \{\mathbf{y}_{i,t}\}\},$$

which yields

$$\begin{aligned} \hat{\mathbf{x}}_{i,t} &= E \left[ \mathbf{X} \middle| \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq t}, \left\{ \underline{\mathbf{z}}_{j,\tau} = Z_{j,\tau} \right\}_{j \in N_i}^{\tau \leq t-1}, \right. \\ &\quad \left. \left\{ \underline{\mathbf{z}}_{j,t-1} \setminus \{Z_{i,t-1} \setminus \underline{\mathbf{y}}_{i,t-1}\} \right\} \right. \\ &\quad \left. = Z_{j,t-1} \setminus \{Z_{i,t-1} \setminus \{\mathbf{y}_{i,t-1}\}\} \right\}_{j \in N_i} \right] \end{aligned} \quad (25)$$

and (25) is also given by (4) and the proof is concluded.  $\square$

We point out that the optimal algorithms can achieve the desired performance provided that the model assumptions are satisfied. If a link or node failure occurs, the algorithms may not achieve the optimal performance. However, once we detect a link failure, we can redesign the algorithms by eliminating this link in the new reconfiguration. Hence, through such strategies, we can increase the robustness of the introduced algorithms.

In the following, we enhance the learning rate further for an arbitrary network topology by exploiting the structure of the fully connected sub-networks that we call as “cell”.

### 3.3. Tree Networks Involving Cell Structures

We define a “cell structure” as a sub-network in which all agents are connected to each other. Intuitively, considering a cell structure as a “single” agent, we can involve the cell (i.e., all the agents in the cell) in the tree such that we can still achieve the distributed-MMSE performance through the disclosure of local estimates (although we may have loops in the cell). We list the features of the cell structures, e.g., seen in Fig. 3, as follows:

- Agents out of a cell can connect to at most one of the agents within that cell.
- A cell structure consists of at least 2 agents.
- An agent can belong to more than one cell.
- Two different agents cannot belong to more than one cell at the same time.
- All of the agents belong to at least a cell in a connected network.
- Each agent has also the knowledge of the cells of the other agents.
- Each agent labels its cells from its own and its first order neighbor’s point of view. As an example, for the agent  $i$ ,  $C_{i,i_1}$  denotes the cell involving both  $i$  and  $i_1$ . Note that if the same cell also includes  $i_2$ ,  $C_{i,i_1} = C_{i,i_2}$ .

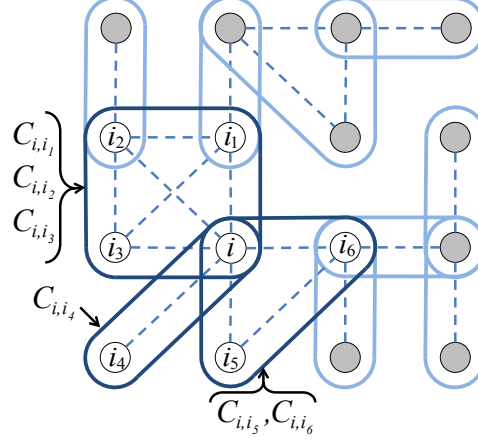


Figure 3: An example tree network involving cell structures.

The following theorem shows that we can achieve the performance of the oracle algorithm over tree networks involving cell structures through the disclosure of the local estimates only.

**Theorem 3.2:** *Consider the distributed estimation framework over tree networks involving cell structures. Then, the distributed-MMSE estimator (3) could also be obtained by (4).*

*Proof:* We point out that  $\hat{Z}_{j,i,t}$  denotes the set of new information received by  $i$  over  $j$  at time  $t$ . Initially, we have  $\hat{Z}_{j,i,0} = Z_{j,0} = \{\mathbf{y}_{j,0}\}$  and the MMSE estimator is also given by (6) over this network topology. Note that the information received by  $j$  at  $t = 1$  is given by  $Z_{j,1} = \left\{ \mathbf{y}_{j,1}, \left\{ \overbrace{\mathbf{y}_{k,0}}^{\hat{Z}_{k,j,0}} \right\}_{k \in N_j}, Z_{j,0} \right\}$ , which yields

$$\begin{aligned} Z_{j,i,1} &= Z_{j,1} \setminus \left\{ Z_{j,0} \cup \bigcup_{k \in C_{i,j}} \{\mathbf{y}_{k,0}\} \right\}, \\ &= Z_{j,1} \setminus \left\{ Z_{j,0} \cup \bigcup_{k \in C_{i,j}} \hat{Z}_{k,j,0} \right\} \end{aligned}$$

and  $\hat{Z}_{j,j,t} = \emptyset$  by definition. Due to the cell structure, we have

$$\begin{aligned} \bigcup_{k \in C_{j,i}} \hat{Z}_{k,j,0} &= \hat{Z}_{i,j,0} \cup \bigcup_{k \in C_{i,j} \setminus j} \hat{Z}_{k,i,0}, \\ &= Z_{i,0} \cup \bigcup_{k \in C_{i,j} \setminus j} Z_{k,0}, \end{aligned}$$

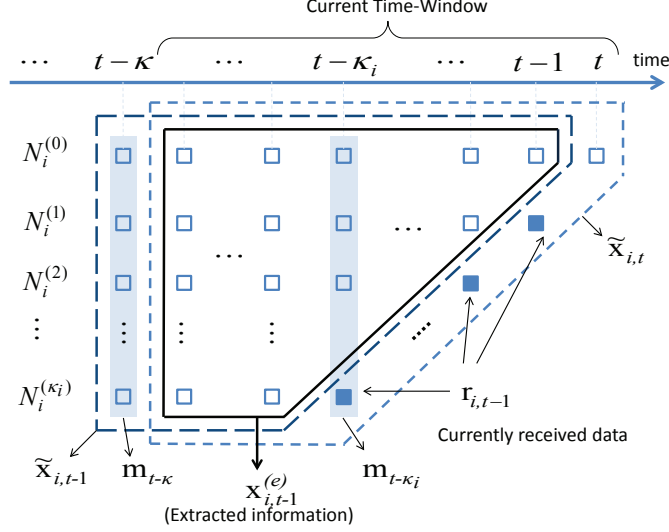


Figure 4: Information aggregation illustration. Small squares represent the information across the agents in the corresponding neighborhood and time.

which leads to

$$\begin{aligned} \hat{\mathbf{x}}_{i,2} &= E \left[ \mathcal{X} \left| \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}^{\tau \leq 2}, \left\{ \mathcal{Z}_{j,0} = Z_{j,0} \right\}_{j \in N_i}, \right. \right. \\ &\quad \left. \left\{ \mathcal{Z}_{j,1} \setminus \left\{ \mathcal{Z}_{j,0} \cup \mathcal{Z}_{i,0} \cup \bigcup_{k \in C_{i,j} \setminus j} \mathcal{Z}_{k,0} \right\} \right\} \right. \\ &\quad \left. \left. = Z_{j,1} \setminus \left\{ Z_{j,0} \cup Z_{i,0} \cup \bigcup_{k \in C_{i,j} \setminus j} Z_{k,0} \right\} \right\}_{j \in N_i} \right]. \end{aligned}$$

We point out that  $C_{i,j} \subset N_i$  and we obtain (13). Correspondingly, for  $t > 0$  we have

$$\hat{Z}_{j,i,t} = Z_{j,t} \setminus \left\{ Z_{j,t-1} \cup \hat{Z}_{i,j,t-1} \cup \bigcup_{k \in C_{i,j} \setminus j} \hat{Z}_{k,i,t-1} \right\}$$

and  $\hat{Z}_{j,i,t}$  is constructible by the sets  $Z_{j,\tau}$  for  $j \in N_i$  and  $\tau \leq t$ . Hence, for  $t > 0$  we obtain (4) and the proof is concluded.  $\square$

Note that we can have loops within the cell structures and still achieve the optimal performance through the diffusion of the local estimates only. In addition to the enhanced estimation rate, this will also increase the robustness of the introduced strategies against the link failures. In the sequel, we provide the sub-optimal extensions of the algorithms for practical applications.



#### 4. Sub-optimal Approaches

Minimization of the cost measure (2) optimally requires relatively excessive computations. We aim to mitigate the problem sub-optimally yet in a computationally efficient approach while achieving comparable performance with the optimal case. As an example, we can approximate the cost measure (2) through time-windowing as follows

$$\tilde{\mathbf{x}}_{i,t} = \min_{\mathbf{x}} E \left[ \left\| \underline{\mathbf{X}} - \mathbf{x} \right\|^2 \middle| \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}_{t-K < \tau \leq t}, \right. \\ \left. \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(1)}, t-K < \tau \leq t-1}, \dots, \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(k_i)}, t-K < \tau \leq t-k_i} \right].$$

In Fig. 4, we illustrate the time-windowing approach over the aggregated observations. We define a memory element  $\mathbf{m}_t$  denoting the expectation of the zero-mean state conditioned on all observation data time-stamped at  $t$ , i.e.,  $\mathbf{m}_t \triangleq E \left[ \underline{\mathbf{X}} \middle| \left\{ \underline{\mathbf{y}}_{i,t} = \mathbf{y}_{i,t} \right\}_{i \in N} \right]$ . We have  $\mathbf{m}_t = \mathbf{M} \mathbf{y}_t$  and  $\mathbf{M} \triangleq \Sigma_x \mathbf{H}^T (\mathbf{H} \Sigma_x \mathbf{H}^T + \Sigma_n)^{-1}$ , where  $\mathbf{H} \triangleq \text{col}\{\mathbf{H}_1, \dots, \mathbf{H}_m\}$ ,  $\Sigma_n \triangleq \text{diag}\{\Sigma_{n_1}, \dots, \Sigma_{n_m}\}$  and  $\mathbf{y}_t \triangleq \text{col}\{\mathbf{y}_{1,t}, \dots, \mathbf{y}_{m,t}\}$ .

Correspondingly,  $\mathbf{x}_{i,t-1}^{(e)}$  represents the extracted information from the estimate  $\tilde{\mathbf{x}}_{i,t-1}$  via the memory element  $\mathbf{m}_{t-K}$ . In particular,  $\mathbf{x}_{i,t-1}^{(e)}$  is defined as follows

$$\mathbf{x}_{i,t-1}^{(e)} \triangleq E \left[ \underline{\mathbf{X}} \middle| \left\{ \underline{\mathbf{y}}_{i,\tau} = \mathbf{y}_{i,\tau} \right\}_{t-K+1 < \tau \leq t-1}, \right. \\ \left. \dots, \left\{ \underline{\mathbf{y}}_{j,\tau} = \mathbf{y}_{j,\tau} \right\}_{j \in N_i^{(k_i)}, t-K+1 < \tau \leq t-k_i-1} \right].$$

This yields  $\mathbf{x}_{i,t-1}^{(e)} = \mathbf{N}_i \mathbf{y}_{i,t-1}^{(c)}$  where  $\mathbf{y}_{i,t-1}^{(c)}$  is the vector collecting all observation data within that window (See Fig. 4) and  $\mathbf{N}_i \triangleq \Sigma_x \hat{\mathbf{H}}_i^T (\hat{\mathbf{H}}_i \Sigma_x \hat{\mathbf{H}}_i^T + \hat{\Sigma}_{n_i})^{-1}$ . We define  $\hat{\mathbf{H}}_i \triangleq \text{col}\{\mathbf{H}_i, \mathbf{H}_i, \mathbf{H}_i^{(1)}, \dots, \mathbf{H}_i, \mathbf{H}_i^{(1)}, \dots, \mathbf{H}_i^{(k_i)}\}$  and  $\mathbf{H}_i^{(j)} \triangleq \text{col}\{\mathbf{H}_{j_1}, \dots, \mathbf{H}_{j_{\pi_i(j)}}\}$ . Correspondingly,  $\hat{\Sigma}_{n_i} \triangleq \text{diag}\{\Sigma_{n_i}, \Sigma_{n_i}, \Sigma_{n_i}^{(1)}, \dots, \Sigma_{n_i}, \Sigma_{n_i}^{(1)}, \dots, \Sigma_{n_i}^{(k_i)}\}$  and  $\Sigma_{n_i}^{(j)} \triangleq \text{diag}\{\Sigma_{n_{j_1}}, \dots, \Sigma_{n_{j_{\pi_i(j)}}}\}$ . Then, after some algebra the estimate is given by  $\tilde{\mathbf{x}}_{i,t-1} = \mathbf{K}_i \mathbf{m}_{t-K} + \mathbf{L}_i \mathbf{x}_{i,t-1}^{(e)}$  where

$$\mathbf{K}_i \triangleq (\mathbf{I} - \mathbf{M} \mathbf{H} \mathbf{N}_i \hat{\mathbf{H}}_i)^{-1} - (\mathbf{I} - \mathbf{N}_i \hat{\mathbf{H}}_i \mathbf{M} \mathbf{H})^{-1} \mathbf{N}_i \hat{\mathbf{H}}_i, \\ \mathbf{L}_i \triangleq (\mathbf{I} - \mathbf{M} \mathbf{H}) (\mathbf{I} - \mathbf{N}_i \hat{\mathbf{H}}_i \mathbf{M} \mathbf{H})^{-1}.$$

Hence, we obtain the sub-optimal distributed online learning (SDOL) algorithm as

$$\tilde{\mathbf{x}}_{i,t} = \mathbf{A}_i \tilde{\mathbf{x}}_{i,t-1} + \mathbf{B}_i \mathbf{y}_{i,t} + \mathbf{C}_i \mathbf{r}_{i,t-1} - \mathbf{D}_i \mathbf{y}_{t-K}, \quad (26)$$

where  $\mathbf{r}_{i,t-1}$  is a vector consisting of currently received observation data from the neighboring agents. The combination matrices defined as

$$\begin{bmatrix} \mathbf{B}_i & \mathbf{C}_i \end{bmatrix} = \Sigma_{x,i} \hat{\mathbf{H}}_i^T (\hat{\mathbf{H}}_i \Sigma_{x,i} \hat{\mathbf{H}}_i^T + \hat{\Sigma}_{n_i})^{-1}, \quad (27)$$

$$\mathbf{A}_i \triangleq (\mathbf{I} - \begin{bmatrix} \mathbf{B}_i & \mathbf{C}_i \end{bmatrix} \hat{\mathbf{H}}_i) \mathbf{L}_i^{-1}, \quad (28)$$

$$\mathbf{D}_i \triangleq \mathbf{A}_i \mathbf{K}_i \mathbf{M}, \quad (29)$$

Table 3: The description of the SDOL algorithm.

---

<b>Algorithm:</b> The SDOL algorithm
<b>Initialization:</b>
$\tilde{\mathbf{x}}_{i,-1} = \mathbf{0}$ for all $i$
$\mathbf{y}_{i,\tau} = \mathbf{0}$ for $\tau < 0$ and all $i$
Calculate combination matrices via (27), (28) and (29).
<b>Update:</b>
Do for $t \geq 0$
For $i = 1$ to $m$ do
Construct $\mathbf{r}_{i,t-1}$ through received $\mathbf{r}_{j,t-1}$ for $j \in N_i^{(1)}$ .
Store $\mathbf{r}_{i,t-1}$ and $\mathbf{y}_{i,t}$ in the memory accordingly.
$\tilde{\mathbf{x}}_{i,t} = \mathbf{A}_i \tilde{\mathbf{x}}_{i,t-1} + \mathbf{B}_i \mathbf{y}_{i,t} + \mathbf{C}_i \mathbf{r}_{i,t-1} - \mathbf{D}_i \mathbf{y}_{t-\kappa}$ .
Erase $\mathbf{y}_{t-\kappa}$ from the memory.
Diffuse $\mathbf{r}_{i,t}$ to the neighboring agents.
End for

---

where  $\Sigma_{x_i} \triangleq (\mathbf{I} - \mathbf{N}_i \hat{\mathbf{H}}_i) \Sigma_x$ . In Table 3, we tabulate the detailed description of the SDOL algorithm. Note that SDOL algorithm is based on the aggregation of information. Correspondingly, we can apply the time-windowing approach to the disclosure of local estimates and formulate a sub-optimal efficient distributed online learning algorithm (SEDOL).

In the following we provide several remarks about the practical implementation of the SDOL algorithm.

**Remark 4.1:**

- We point out that in the update (26) the matrices  $\mathbf{A}_i$ ,  $\mathbf{B}_i$ ,  $\mathbf{C}_i$ , and  $\mathbf{D}_i$  are independent from the data and they are time invariant. Hence, they can be pre-computed and installed onto the agents. Then, the SDOL algorithm basically takes the linear average of the previous estimate, the current measurement and received data, and observation data at time  $t - \kappa$ .
- Different from the conventional approaches, the SDOL algorithm requires to memorize previous observations. Note that if  $q < p/m$ , storing observation data individually rather than a linear combination, i.e.,  $\mathbf{m}_t$ , might be more efficient in terms of memory usage.
- The computational complexity of the SDOL algorithm is  $O(p^2 + 2pmq)$  (or say  $O(p^2)$  for  $q \ll p/m$ ) and the algorithm requires disclosure of a data vector with  $(m-1) \times q$  dimensions.
- Finally, the steady-state MSE of the SDOL algorithm is given by

$$\text{MSE} = \text{Tr} \{ \mathbf{A}_i \mathbf{L}_i \Sigma_{x_i} \}.$$

In the sequel, we also provide illustrative simulations showing the enhanced tracking performance due to the proposed algorithm over several distributed network scenarios.

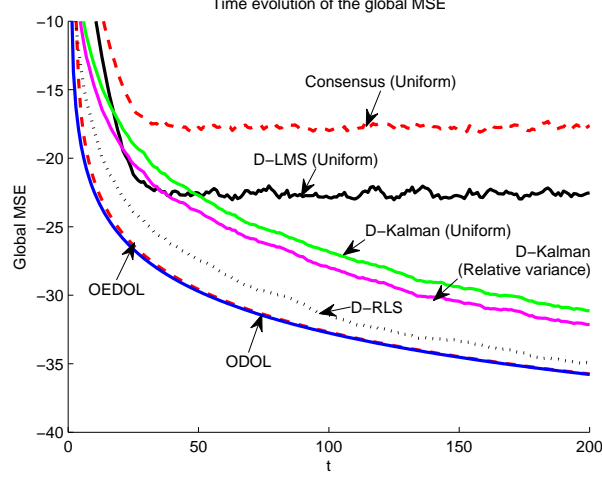


Figure 5: Comparison of the global MSE of the algorithms over an arbitrary network where the agents observe the same noise statistics.

## 5. Illustrative Examples

In this section, we examine the performance of the introduced strategies under different scenarios. Consider a network of  $m = 20$  agents where each agent  $i$  observes an underlying state  $\mathbf{x} \in \mathbb{R}^p$  through  $y_{i,t} = \mathbf{h}_i^T \mathbf{x} + n_{i,t}$ . The unknown state  $\mathbf{x}$  is a realization of standard-normal distribution. The observation noise is a zero-mean white Gaussian random process. For comparison, we choose  $\mathbf{h}_i \in \mathbb{R}^p$  randomly from a standard-normal distribution. We define a global MSE performance measure over a network as  $1/m \sum_{i=1}^m \text{MSE}(i)$ , where  $\text{MSE}(i)$  represents the MSE of the  $i$ th agent.

We compare the performance of the introduced algorithms with the diffusion implementation of Kalman updates (D-Kalman) [9], diffusion least mean square (D-LMS) [8], diffusion recursive least squares (D-RLS) [35], and consensus algorithms [36]. We utilize several combination rules. In the uniform combination rule [20], the combination weights given by agent  $i$  for neighbor  $j$  are chosen as

$$\lambda_{i,j} = \begin{cases} 1/(\pi_i + 1) & \text{if } j \in N_i \cup \{i\} \\ 0 & \text{otherwise} \end{cases}$$

since  $N_i$  excludes  $i$ . Correspondingly, in the relative variance rule [37, 38],  $\lambda_{i,j}$  is given by

$$\lambda_{i,j} = \begin{cases} \frac{\sigma_{n_j}^{-2}}{\sum_{k \in N_i \cup \{i\}} \sigma_{n_k}^{-2}} & \text{if } j \in N_i \cup \{i\} \\ 0 & \text{otherwise.} \end{cases}$$

In the D-LMS and consensus algorithms, we set  $\mu = 0.1$ . In the D-RLS algorithm, we use the Laplacian combination rule [39] for the incremental update and the Metropolis combination rule for the spatial update.

In Figs. 5 and 6, we compare the time evolution of the global MSE of all algorithms for space-invariant and space-variant noise statistics, respectively for  $p = 1$ . We set the standard deviations

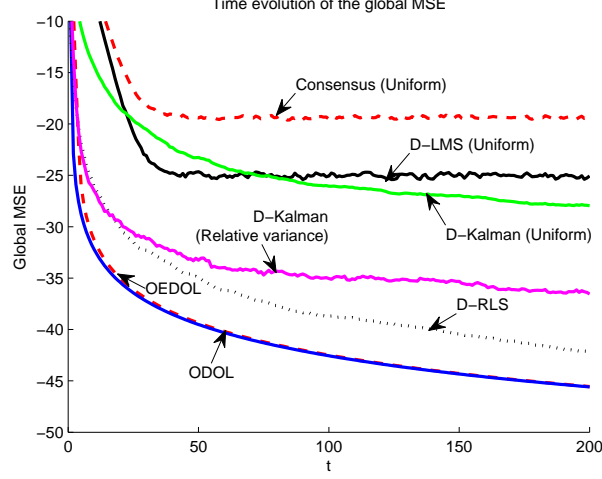


Figure 6: Comparison of the global MSE of the algorithms over an arbitrary network where the agents observe different noise statistics.

of the noise parameters from a folded standard normal distribution in the space-variant case. The network topology is chosen arbitrarily and we can readily construct a corresponding spanning tree of that network. As an example, we can construct the spanning tree based on the paths from the most connected agent  $i$  to the others, i.e.,  $i = \arg \max_j \pi_j$ , and then we eliminate the multi-paths [27]. The relative-variance cooperation rule that considers the noise statistics to adjust the combination weights performs better than the uniform combination rule for space-variant noise profiles across the network. In Fig. 7, we compare the performance of the proposed algorithms for relatively large state vector, i.e.,  $p = 10$ , and space variant noise profiles. In Figs. 5, 6 and 7 we observe that the proposed algorithms achieve superior performance compared to the other algorithms through the optimal weight selection for the MMSE performance. Additionally, the performance of the OEDOL and ODOL algorithms are close to each other even though the OEDOL algorithm operates over the corresponding spanning tree.

In order to examine the influence of the network topology on the MMSE performance, we provide Fig. 8. The fully connected network (or the centralized network) achieves the best possible connection among the agents whereas the line network yields the least possible connection. Over a star network, the furthest distance between the agents is 2 and a single agent, i.e., the so-called pseudo-centralized agent, can access to the information of its neighbors directly. Notably, the star and line networks are the spanning tree of the fully connected network. We observe that the MMSE over the star network is close to the MMSE over the fully connected network. Additionally, the MMSE over even the line network converges to the MMSE over the fully connected network asymptotically.

For practical implementations, we propose the SDOL algorithm. In Fig. 9, we examine the influence of the time-windowing depth on the global MSE performance of the SDOL algorithm. We use the SDOL algorithms with time-windowing depths  $\kappa = 20$  and  $\kappa = 50$ . The time evolution of the MSE performance differs from the MMSE behavior because the SDOL algorithms inherently assume that even at initial time, i.e.,  $t = 0$ , agents have legitimate observations from

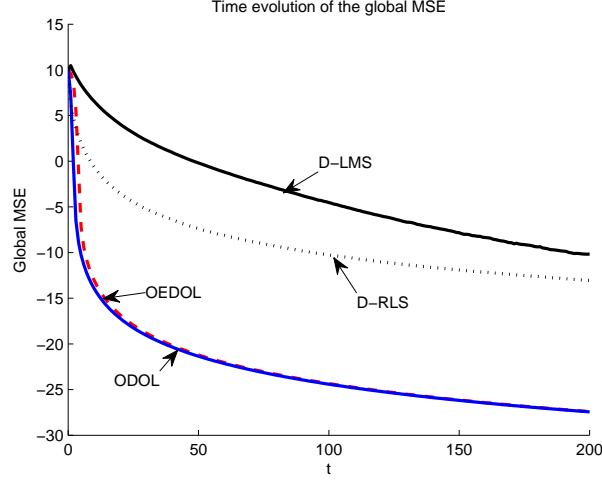


Figure 7: MSE performances of the introduced algorithms for  $p = 10$ .

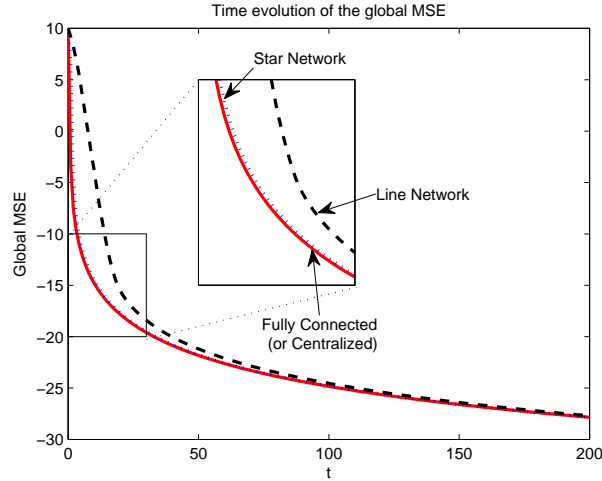


Figure 8: Comparison of the global MMSE curves over fully connected, star and line networks.

$t = \kappa$  to  $t = 0$ . By excluding this data in time through sliding time-window, the SDOL algorithms eventually reach the best performance at the corresponding time-windowing depths and maintain this performance afterwards.

Finally, we evaluate the MSE performance of the SDOL algorithm for abruptly changing state. Assume that the state vector changes at time  $t = 400$  abruptly. In particular, we have chosen a new realization from the standard normal distribution. In Fig. 10, we compare the SDOL algorithms with the D-LMS algorithm. Note that we set the step size of the D-LMS algorithm as  $\mu = 0.025$  so that at steady state (if there were no state change) the D-LMS and SDOL with

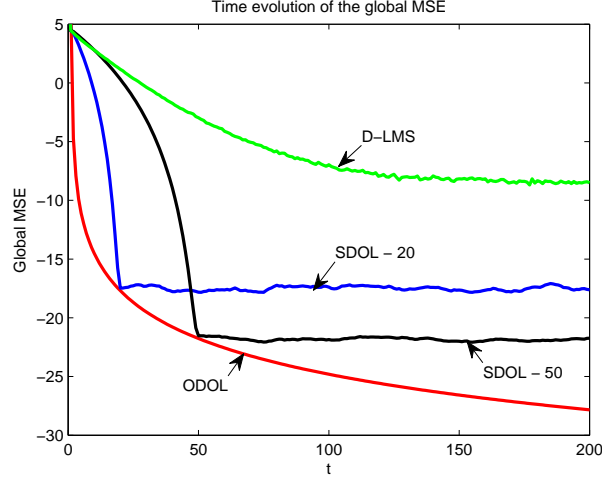


Figure 9: Influence of the time-windowing on the global MSE performance.

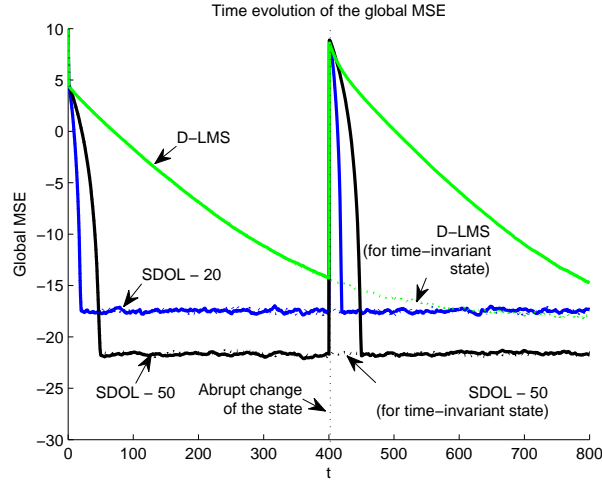


Figure 10: Tracking performance of the SDOL algorithms if the state changes abruptly.

$\kappa = 20$  algorithms achieve the same MSE performance for fair tracking performance comparison. The sliding time-window provides relative robustness against abrupt state changes due to the exclusion of the effect of the previous observations. Hence, the SDOL algorithm can also provide enhance tracking performance in the practical applications.

## 6. Conclusion

The distributed algorithms have attracted significant attention due to their wide spread applicability to highly complex structures from biological systems to social and economical networks.

However, there are still challenges for disclosure and utilization of information among agents. We introduce the ODOL algorithm achieving the MMSE performance for Gaussian state and noise statistics and the LMMSE performance for arbitrary statistical profiles. The ODOL algorithm has consensus-like iterations for streaming data over an arbitrary network topology via the aggregation of information at each agent rather than propagation of information through the disclosure of local estimates. Importantly, the disclosure of the local estimate is sufficient only over the certain network topologies, e.g., over the introduced tree network involving cell structures. Moreover, the aggregation of information based approach can require excessive communication load. Therefore, in order to reduce the communication load, we exploit the network structures. We introduce the OEDOL algorithm that achieves the MMSE performance through the disclosure of local estimate over tree networks. We also observe that the MSE performance of the ODOL and OEDOL algorithms are close to each other over even a highly connected arbitrary network while the OEDOL provides reduced communication load in general. Finally, we introduce time-windowing approach for practical applications due to the reduced complexity. The sub-optimal approaches possess consensus-like iterations with time-invariant combination weights that should be calculated only once. Additionally, the sub-optimal approaches provide enhanced tracking performance against changing state due to the sliding of time-window.

## 7. Acknowledgment

This work is in part supported by the Outstanding Researcher Programme of Turkish Academy of Sciences and TUBITAK projects, Contract no: 112E161 and 113E517.

## References

### References

- [1] M. K. Banavar, J. J. Zhang, B. Chakraborty, H. Kwon, Y. Li, H. Jiang, A. Spanias, C. Tepedelenioglu, C. Chakrabarti, A. P.-Suppappola, An overview of recent advances on distributed and agile sensing algorithms and implementation, *Digital Signal Processing* 39 (2015) 1–14.
- [2] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, Z. J. Towfic, Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior, *IEEE Signal Processing Magazine* 30 (3) (2013) 155–171.
- [3] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, J. Arenas-Garcia, M. T. M. Silva, Distributed estimation in diffusion networks using affine least-squares combiners, *Digital Signal Processing* 36 (2015) 1–14.
- [4] V. Krishnamurthy, Decentralized activation in sensor networks-global games and adaptive filtering games, *Digital Signal Processing* 21 (5) (2011) 638–647.
- [5] A. Jadbabaie, P. Molavi, A. Sandroni, A. Tahbaz-Salehi, Non-Bayesian social learning, *Games and Economic Behavior* 76 (1) (2012) 210–225.
- [6] D. Acemoglu, A. Ozdaglar, Dynamic games and applications, *Opinion dynamics and learning in social networks* 1 (1) (2011) 3–49.
- [7] M. Jackson, *Social and Economic Networks*, Princeton University Press, Princeton, N.J., 2008.
- [8] C. G. Lopes, A. H. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis, *IEEE Transactions on Signal Processing* 56 (7) (2008) 3122–3136.
- [9] F. S. Cattivelli, A. H. Sayed, Diffusion strategies for distributed Kalman filtering and smoothing, *IEEE Transactions on Automatic Control* 55 (9) (2010) 2069–2084.
- [10] S. Shahrampour, A. Rakhlin, A. Jadbabaie, Online learning of dynamic parameters in social networks, in: *Neural Information Processing Systems*, 2013, pp. 2013–2021.
- [11] I. Schizas, G. Mateos, G. Giannakis, Distributed LMS for consensus-based in-network adaptive processing, *IEEE Transactions on Signal Processing* 57 (6) (2009) 2365–2382.
- [12] M. O. Sayin, S. S. Kozat, Compressive diffusion strategies over distributed networks for reduced communication load, *IEEE Transactions on Signal Processing* 62 (20) (2014) 5308–5323.



- [13] J. I. Steinfeld, J. S. Francisco, W. L. Hase, Chemical Kinetics and Dynamics, Pearson, 1999.
- [14] M. O. Sayin, S. S. Kozat, Single bit and reduced dimension diffusion strategies over distributed networks, *IEEE Signal Processing Letters* 20 (10) (2013) 976–979.
- [15] R. Olfati-Saber, Distributed kalman filtering for sensor networks, in: *Proceedings of 46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.
- [16] A. Ribeiro, G. B. Giannakis, S. I. Roumeliotis, SOI: Distributed Kalman filtering with low-cost communications using the sign of innovations, *IEEE Transactions on Signal Processing* 54 (12) (2006) 4782–4795.
- [17] M. S. Mahmoud, H. M. Khalid, Distributed Kalman filtering: a bibliographic review, *IET Control Theory and Applications* 7 (4) (2013) 483–501.
- [18] N. Assimakis, M. Adam, M. Koziri, S. Voliotis, K. Asimakis, Optimal decentralized Kalman filter and Lainiotis filter, *Digital Signal Processing* 23 (1) (2013) 442–452.
- [19] R. J. Aumann, Agreeing to disagree, *The Annals of Statistics* 4 (6) (1976) 1236–1239.
- [20] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, J. N. Tsitsiklis, Convergence in multiagent coordination, consensus, and flocking, in: *Proceedings of 44th IEEE Conference on Decision and Control (CDC-ECC)*, 2005, pp. 2996–3000.
- [21] D. S. Scherber, H. C. Papadopoulos, Locally constructed algorithms for distributed computations in ad-hoc networks, in: *Proceedings of Information Processing Sensor Networks (IPSN)*, 2004, pp. 11–19.
- [22] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of state calculations by fast computing machines, *The Journal of Chemical Physics* 21 (6) (1953) 1087–1092.
- [23] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, New York, 2003.
- [24] M. H. DeGroot, Reaching a consensus, *Journal of the American Statistical Association* 69 (345) (1974) 118–121.
- [25] B. D. O. Anderson, J. B. Moore, *Optimal Filtering*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1979.
- [26] S. Skiena, *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Addison-Wesley, Reading, MA, 1990.
- [27] B. Y. Wu, K.-M. Chao, *Spanning Trees and Optimization Problems (Discrete Mathematics and Its Applications)*, CRC Press, New York, 2004.
- [28] R. G. Gallager, P. A. Humblet, P. M. Spira, A distributed algorithm for minimum-weight spanning trees, *ACM Transactions on Programming Languages and Systems* 5 (1) (1983) 66–71.
- [29] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*, SIAM, Philadelphia, PA, 2000.
- [30] M. Elkin, Unconditional lower bounds on the time-approximation tradeoffs for the distributed minimum spanning tree problem, in: *Proceedings of 36th ACM Symposium on Theory of Computing (STOC)*, 2004, pp. 331–340.
- [31] M. Khan, G. Pandurangan, V. S. A. Kumar, Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks, *IEEE Transactions on Parallel and Distributed Systems* 20 (1) (2009) 124–139.
- [32] J. Chen, A. H. Sayed, Optimal combination rules for adaptation and learning over networks, in: *Proceedings of 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012, pp. 1535–1542.
- [33] J. Chen, A. H. Sayed, On the learning behavior of adaptive networks - Part II: transient analysis, *IEEE Transactions on Information Theory* 61 (6).
- [34] J. Chen, A. H. Sayed, On the learning behavior of adaptive networks - Part II: performance analysis, *IEEE Transactions on Information Theory* 61 (6).
- [35] F. S. Cattivelli, C. G. Lopes, A. H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks, *IEEE Transactions on Signal Processing* 56 (5) (2008) 1865–1877.
- [36] S. Y. Tu, A. H. Sayed, Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks, *IEEE Transactions on Signal Processing* 60 (12) (2012) 6217–6234.
- [37] F. S. Cattivelli, A. H. Sayed, Diffusion LMS for distributed estimation over adaptive networks, *IEEE Transactions on Signal Processing* 58 (3) (2010) 1035–1048.
- [38] S.-Y. Tu, A. H. Sayed, Optimal combination rules for adaptation and learning over networks, in: *Proceedings of 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2011, pp. 317–320.
- [39] R. Olfati-Saber, R. M. Murray, Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control* 49 (9) (2004) 1520–1533.